The PDE surface method in higher dimensions

Woodland, Alan; Ugail, Hassan; Labrosse, Frédéric

Publication date:
2007

Citation for published version (APA):
Woodland, A., Ugail, H., & Labrosse, F. (2007). *The PDE surface method in higher dimensions*. British Machine Vision Association. http://hdl.handle.net/2160/3096

The PDE surface method in higher dimensions

A. Woodland[1], H. Ugail[2], F. Labrosse[1]

[1]Department of Computer Science, University of Wales, Aberystwyth

[2]School of Informatics, University of Bradford

June 6, 2008

Abstract

This paper presents a method to extend PDE surfaces to high dimensional spaces. We review a common existing analytic solution, and show how it can be used straightforwardly to increase the dimension of the space the surface is embedded within. We then further develop a numerical scheme suitable for increasing the number of variables that parametrise the surface, and investigate some of the properties of this solution with a view to future work.

1 Introduction

The concept of *image manifolds* is one that has become increasingly common in vision [7, 1, 5], and recently to some extent in graphics literature. Images, which can be considered points in a high-dimensional *image space*, of an object or environment undergoing a series of transformations (which are typically parameterised in some way) are not randomly placed in image space. Instead under certain circumstances these can be approximated as n-manifolds, where n is the number of independent variables controlling the image transformations. This mapping between the high-dimensional image space, and a lower dimensional space has been particularly useful in the area of vision [8], however developing efficient in-memory representations of image manifolds has proven hard, and many researchers have used techniques like PCA to simplify the representations. This is acceptable if the intended use is recognition of images, however for the purpose of synthesising new images it proves to lose too many of the details that make images appear realistic to a human observer. A potential solution to this problem is to develop exisiting surface representation techniques to model these structures in image space directly, thereby representing all of the images that make up the manifold.

The method of representing surfaces using partial differential equations, referred to as the *PDE surface method* henceforth, has been proposed [2, 11] as an alternative to methods such as NURBS [9] in a typical CAD application. In this paper we look at extending this method to model surfaces (or more generally manifolds) which are embedded in much higher dimensional spaces than required for a CAD application, making it suitable for modelling image manifolds. Additionally we look at increasing the number of parameters used to control the manifold, which can be more than the typical (u, v) parametrisation

of a surface used in a CAD application. One such existing example of using the PDE surface method to model a volume is presented in [6].

This paper is split into five sections. In Section 2 we provide an overview of the problem, and a common analytic solution which is then straightforwardly extended to spaces of much higher dimension. Section 3 discusses approaches to increasing the dimension of the parameter space, which requires a numerical solution to the resulting PDE. This work is conducted with a view to the specific application of *image manifolds*, and some early results are presented and briefly discussed in Sections 4 and 5 respectively, along with desription of some future work we intend to undertake as a result of this work.

2 Increasing the embedding space

There have been many different surface modelling techniques proposed in the past, in particular parametric surface modelling.

A parametric surface $\underline{X}(u,v)$ can be expressed

$$\underline{X}(u,v) = (x(u,v), y(u,v), z(u,v)) \qquad x,y,z \in \mathbb{R} \tag{1}$$

We now assume our surface to be periodic in v, and restrict it to the finite domain $\Omega = \{u,v : 0 \leq u \leq 1; 0 \leq v \leq 2\pi\}$. We therefore have that

$$\left(\frac{\partial^2}{\partial u^2} + \alpha^2 \frac{\partial^2}{\partial v^2}\right)^2 \underline{X}(u,v) = 0, \tag{2}$$

where α is a "smoothing parameter", which controls the length over which the boundary conditions influence the interior of the surface. This equation is often referred to as the *biharmonic*

Existing work using the PDE surface method [11] has used the method of separation of variables [4] to find an analytic solution to the PDE above, for which a computationally efficient implementation is also possible. We also assume now that sufficient boundary conditions have been imposed upon the surface. The basic solution to this is now outlined.

Given that Eqn. (2) is elliptic, the boundary conditions continuous and closed we use the following general solution:

$$\underline{X}(u,v) = \underline{A}_0(u) + \sum_{n=1}^{\infty} \left[\underline{A}_n(u)\cos(nv) + \underline{B}_n(u)\sin(nv)\right], \tag{3}$$

where $\underline{A}_0$, $\underline{A}_n$ and $\underline{B}_n$ are vector valued functions as follows:

$$\underline{A}_0 = \underline{a}_{00} + \underline{a}_{01}u + \underline{a}_{02}u^2 + \underline{a}_{03}u^3 \tag{4}$$

$$\underline{A}_n = \underline{a}_{n1}e^{\alpha n u} + \underline{a}_{n2}ue^{\alpha n u} + \underline{a}_{n3}e^{-\alpha n u} + \underline{a}_{n4}ue^{-\alpha n u} \tag{5}$$

$$\underline{B}_n = \underline{b}_{n1}e^{\alpha n u} + \underline{b}_{n2}ue^{\alpha n u} + \underline{b}_{n3}e^{-\alpha n u} + \underline{b}_{n4}ue^{-\alpha n u} \tag{6}$$

By using Fourier analysis the boundary conditions can be written in a form whereby the values of the constants in the general solution can be calculated.

In practise not all boundary conditions have a Fourier series as simple as that of a circle or ellipse, and for that reason the Fourier series is commonly truncated at some given value of n, typically ([11]) $N = 6$. A remainder term can be used

to approximate the higher order components of the boundary condition that would otherwise be lost:

$$\underline{X}(u,v) = \underline{A}_0(u) + \sum_{n=1}^{N} [\underline{A}_n(u)\cos(nv) + \underline{B}_n(u)\sin(nv)] + \underline{R}(u,v), \qquad (7)$$

where $\underline{R}(u,v)$ is usually chosen to be of the form

$$\underline{R}(u,v) = r_0 e^{\omega u} + r_1 e^{-\omega u} + r_2 u e^{\omega u} + r_3 u e^{-\omega u}, \qquad (8)$$

where $\underline{R}_0(v), \ldots, \underline{R}_3(v)$ are calculated as the difference between the real boundary conditions, and the approximation of them. Choosing $\omega = \alpha(N+1)$ gives a decay rate similar to the decay rate of the actual solution given that the mode $(N+1)$ is the dominant mode of the modes that have been truncated.

Given the solution outlined above it is straight forward to increase the dimensionality of the space in which the surface is embedded; one simply has to increase the dimensionality of the vector-valued functions $\underline{A}_0$, $\underline{A}_n$ and $\underline{B}_n$. This in turn simply requires one to specify boundary conditions for the PDE in each additional dimension, and solve the corresponding linear system of equations.

3 The parameter space

Extending the surface to be controlled by more than 2 parameters (an n-manifold) is a more challenging problem — the solution outlined in Section 2 is no longer applicable, and we must turn now to a numerical solution [10].

In order to solve for the case with more than two parameters we first derive a numerical scheme for the solution of the lower order case, namely Laplace's equation (Eqn. (9)) which is simpler than we previously solved analytically. A similar scheme for Eqn. (2) is straightforward to derive using the same method.

$$\left(\frac{\partial^2}{\partial u^2} + \frac{\partial^2}{\partial v^2}\right) \underline{X}(u,v) = 0 \qquad (9)$$

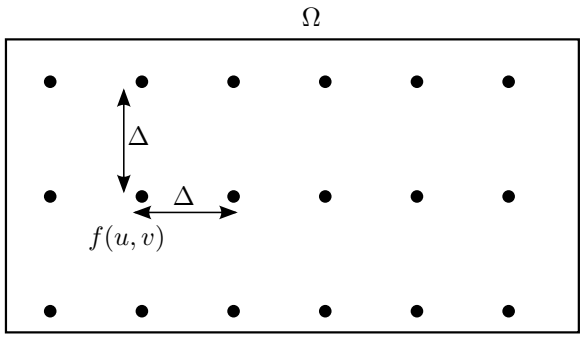


Figure 1: Discretisation of the domain Ω

Firstly we discretise our domain Ω and produce a finite grid (Fig. 1) that represents it. Then we seek a discrete approximation in terms of the grid of the various terms involved, firstly wrt. u. We assume here a uniform grid to simplify

the problem, although this need not always be the case. In this case we have used a centred difference scheme, because it is a more accurate approximation of the derivatives.

$$\frac{\partial}{\partial u} \approx \frac{1}{\Delta}\left[f(u+\frac{\Delta}{2},v) - f(u-\frac{\Delta}{2},v)\right] \tag{10}$$

A discrete approximation of $\frac{\partial}{\partial v}$ is produced similarly. We can now proceed further to produce discrete approximations on the gird of higher order terms:

$$\frac{\partial^2}{\partial u^2} \approx \frac{1}{\Delta^2}\left[f(u+\Delta,v) - 2f(u,v) + f(u-\Delta,v)\right], \tag{11}$$

and similarly

$$\frac{\partial^2}{\partial v^2} \approx \frac{1}{\Delta^2}\left[f(u,v+\Delta) - 2f(u,v) + f(u,v-\Delta)\right], \tag{12}$$

By substituting Eqn. (11) and (12) into Eqn. (9) and simplifying slightly we get:

$$\frac{1}{\Delta^2}\left[f(u+\Delta,v) + f(u-\Delta,v)f(u,v+\Delta) + f(u,v-\Delta) - 4f(u,v)\right] = 0. \tag{13}$$

The resulting schemes for 2-D and 3-D versions of 2^{nd}, 4^{th} and 6^{th} order equations are illustrated in Figure 3. Similar schemes were produced for 4 and 5 dimensional versions, but for obvious practical reasons they are not illustrated.

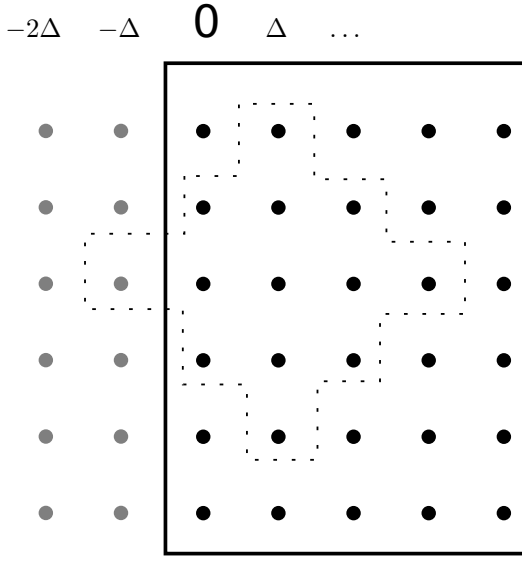


Figure 2: At the edges of the grid 'imaginary' points are calculated to fill in for the higher order schemes

We then apply this scheme to the entire grid, filling in the specified boundary values at the boundaries. This gives a system of linear equations.

$$Ax = B. \tag{14}$$

Notice that in the case of the higher order schemes near the edge of our grid we will need to compute values for points outside of the domain itself; in order to achieve the desired continuity right up to the boundary. This is because the higher order schemes express a given point in terms of more than just its immediate neighbours, yet we seek a solution to these equations right up to the edge of the domain where our positional boundary conditions have been defined. These points are often referred to as *ghost points*, illustrated in Fig. 2.

This scheme produces a system of equations of the form:

$$\begin{bmatrix} \ddots & & & & & & & & & & \\ & \ddots & & & & & & & & & \\ & & \ddots & & & & & & & & \\ & & & \ddots & & & & & & & \\ & 1 & \dots & 1 & -4 & 1 & \dots & 1 & & & \\ & & & & & \ddots & & & & & \\ & & & & & & \ddots & & & & \\ & & & & & & & \ddots & & & \\ & & & & & & & & \ddots & & \\ & & & & & & & & & \ddots & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ x_{n^2-1} \\ x_{n^2} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ b_{n^2-1} \\ b_{n^2} \end{bmatrix} \tag{15}$$

Having produced a suitable linear system of equations that approximates the solution to our given PDE we now seek an efficient solution of that system. In all but the most trivial of cases any direct solution (e.g. inversion, or via some decomposition) is not feasible. Therefore we turn to an iterative approach, for which the reader is referred to the texts [12, 3].

4 Results

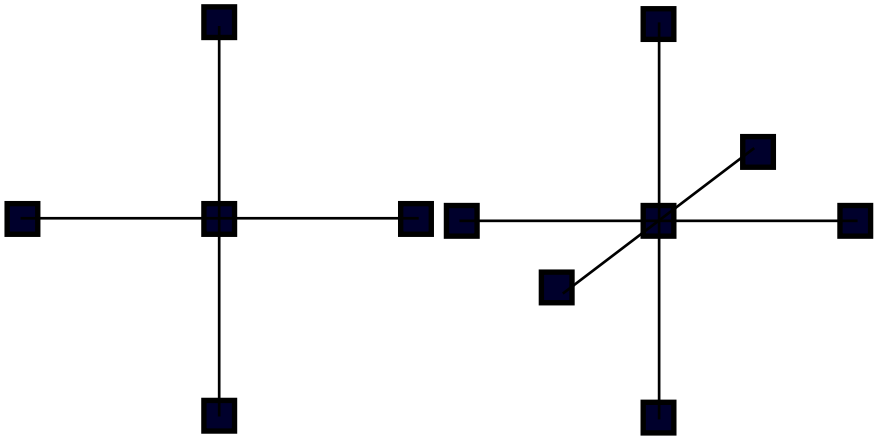
For the purposes of testing, a simple example application, Figure. 4 was produced to demonstrate the two different solutions running side by side for the biharmonic (Eqn. 2). In both cases the boundary conditions specified were identical. The implementation used here utilises a simple multigrid solver, with symmetric Gauss-Seidel relaxations performed at each stage.

Some example computation times with the schemes we produced are presented in Table 1. Note that in this table the 4^{th} order 3-D scheme we use is the same as in [6]. This implementation has not been particularly aggressively optimised yet, and in practice significant performance gains can most probably be made.

In Figure 5 the computational cost of increasing the dimension of the space in which the surface is embedded is investigated. The time is clearly proportional to the dimensionality of the space.

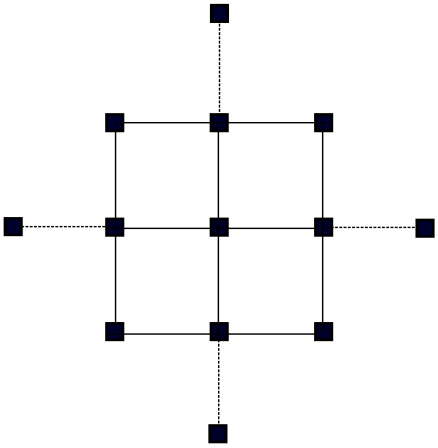
5 Conclusion

The original motivation for this work was that of modelling image manifolds. The results presented here demonstrate the feasibility of this idea from a purely

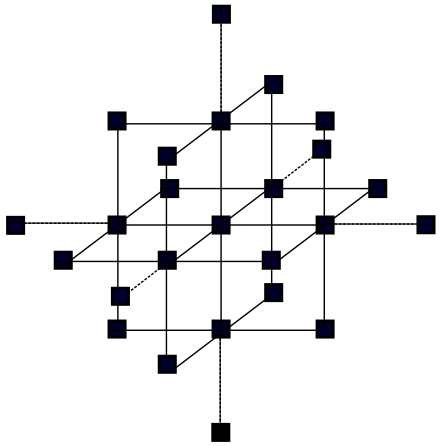


(a) 2^{nd} order 2-D scheme

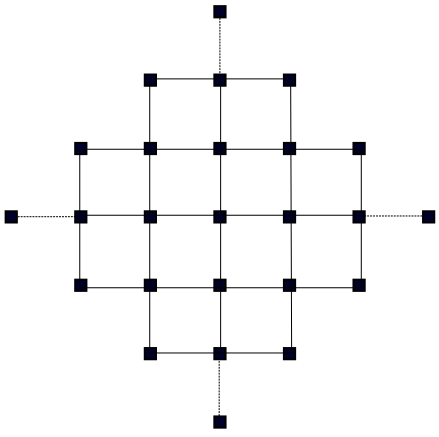
(b) 2^{nd} order 3-D scheme



(c) 4^{th} order 2-D scheme

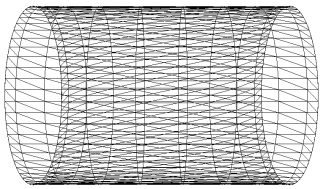


(d) 4^{th} order 3-D scheme

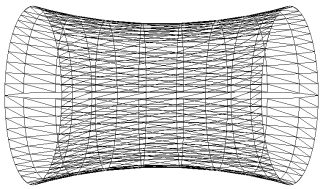


(e) 6^{th} order 2-D scheme

Figure 3: Visual illustrations of several of the schemes we used.



(a) Biharmonic example (analytic)



(b) Biharmonic example (numeric)

Figure 4: Side by side visual comparison of the results of the two different methods for a simple 2-D biharmonic. The numerical solution has a relatively high maximum residual error ($\epsilon_{\max} = 0.01$), but only required a small number of iterations. If we continued until a smaller maximum residual error is obtained, then the numeric solution does converge to be the closer to the analytic one.

Method/scheme	Computation time per dimension	Max Residual error (Iterations)
2^{nd} Order 2-D numeric	0.010s	0.00854 (13)
2^{nd} Order 3-D numeric	0.320s	0.00949 (15)
2^{nd} Order 4-D numeric	29.250s	0.0118 (17)
4^{th} Order 2-D numeric	0.020s	0.0098 (90)
4^{th} Order 3-D numeric	1.180s	0.00998 (137)
4^{th} Order 4-D numeric	197.430s	0.00985 (180)
6^{th} Order 2-D numeric	0.090s	0.00999 (544)
6^{th} Order 3-D numeric	7.030s	0.0118 (1000)
6^{th} Order 4-D numeric	1034.530s	0.0442 (1000)
6^{th} Order 6-D numeric	*dnf*	*dnf*

Table 1: Computational results from numerical schemes. In all cases a 20^d grid was used. The final scheme failed to run because of memory requirements.

computational point of view, and have opened a lot of potential avenues for further work. Work is therefore now progressing on a more detailed study of the application of the techniques developed here, along with several other surface modelling techinques, to the specific problem of modelling image manifolds.

The multigrid solver we used for the results presented here was relatively simplistic, and does not exploit the potential for speed increases by parallel solving of the system of equations. In the future however with large image manifolds it is anticipated that it will be necessary to use parallel solvers on large clusters, and there is a great potential for further speed increases in other areas.

Another interesting avenue for further work is the possibility of using the multigrid solutions to form the basis of a *dynamic level of detail* mechanism for the rendering of PDE surfaces analagous to current dynamic level of detail techniques used when rendering triangle meshes. This could also potentially be

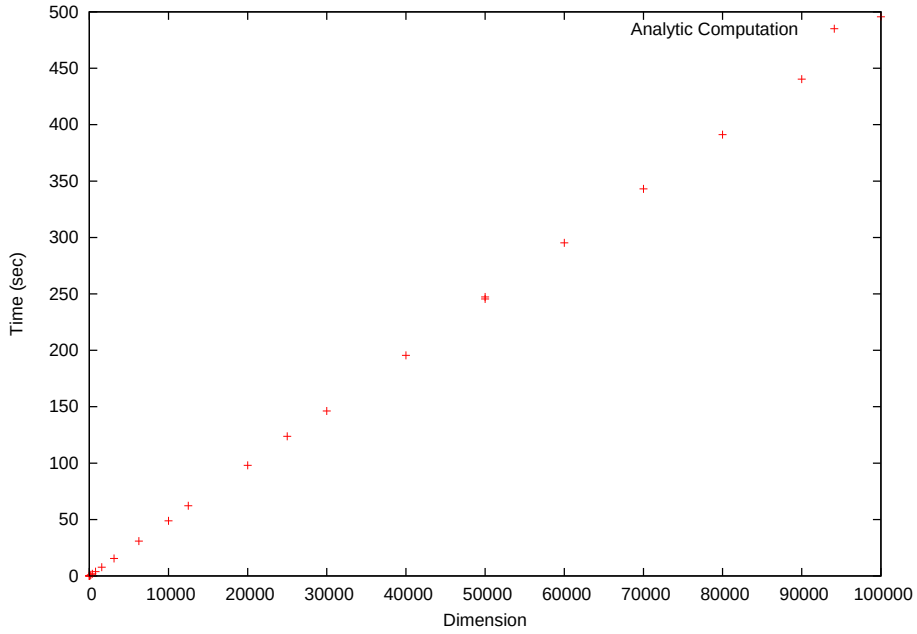


Figure 5: Computational results from analytic solution. As we increase the dimension of the space the surface is embeded in the computational time increases proportionally.

applied when rendering images using image manifolds.

Acknowledgements

The authors would like to thank the VVG network for the grant which made this work possible.

References

[1] M. Bichsel and A. P. Pentland. Human face recognition and the face image set's topology. *CVGIP: Image Underst.*, 59(2):254–261, 1994.

[2] M. I. G. Bloor and M. J. Wilson. Using partial differential equations to generate free-form surfaces. *Comput. Aided Des.*, 22(4):202–212, 1990.

[3] William L. Briggs, Van Emden Henson, and Steve F. McCormick. *A Multigrid Tutorial*. Other Titles in Applied Mathematics. SIAM, second edition, 2000.

[4] Ruel V. Churchill and James Ward Brown. *Fourier Series and Boundary Value Problems*. McGraw-Hill, Inc., 1978.

[5] David L. Donoho and Carrie Grimes. Image manifolds which are isometric to Euclidean space. *Journal of Mathematical Imaging and Vision*, 23(1):5–24, 2005.

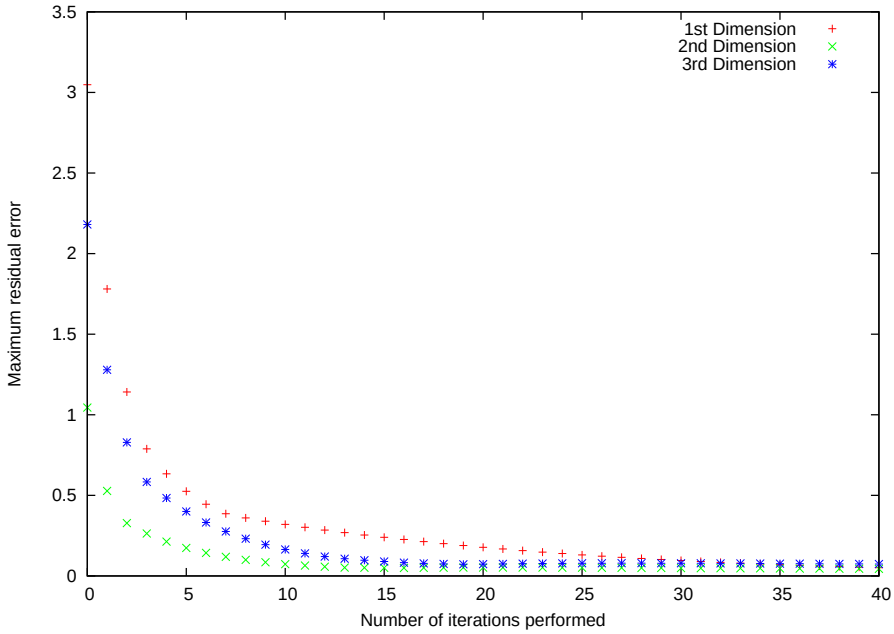


Figure 6: Maximum residual error vs. iterations with the numerical biharmonic scheme as used in Figure 4(b).

[6] Haixia Du and Hong Qin. Free-form geometric modeling by integrating parametric and implicit PDEs. *IEEE Transactions on Visualization and Computer Graphics*, 13(3):549–561, 2007.

[7] Haw-minn Lu, Yeshaiahu Fainman, and Robert Hecht-Nielsen. Image manifolds. In Nasser M. Nasrabadi and Aggelos K. Katsaggelos, editors, *Applications of Artificial Neural Networks in Image Processing III*, volume 3307, pages 52–63. SPIE, 1998.

[8] S. Naya, S. Nene, and H. Murase. Subspace methods for robot vision. *IEEE Transactions on Robotics and Automation*, 12:750–758, 1996.

[9] Les Piegl and Wayne Tiller. *The NURBS book*. Springer, 2nd edition, 1997.

[10] J.W. Thomas. *Numerical Partial Differential Equations: Finite Difference Methods*, volume 22 of *Texts in Applied Mathematics*. Springer, second edition, 1995.

[11] Hassan Ugail, Malcolm I. G. Bloor, and Michael J. Wilson. Techniques for interactive design using the PDE method. *ACM Trans. Graph.*, 18(2):195–212, 1999.

[12] R. S. Varga. *Matrix Iterative Analysis*. Englewood Cliffs, NJ, USA, 1962.